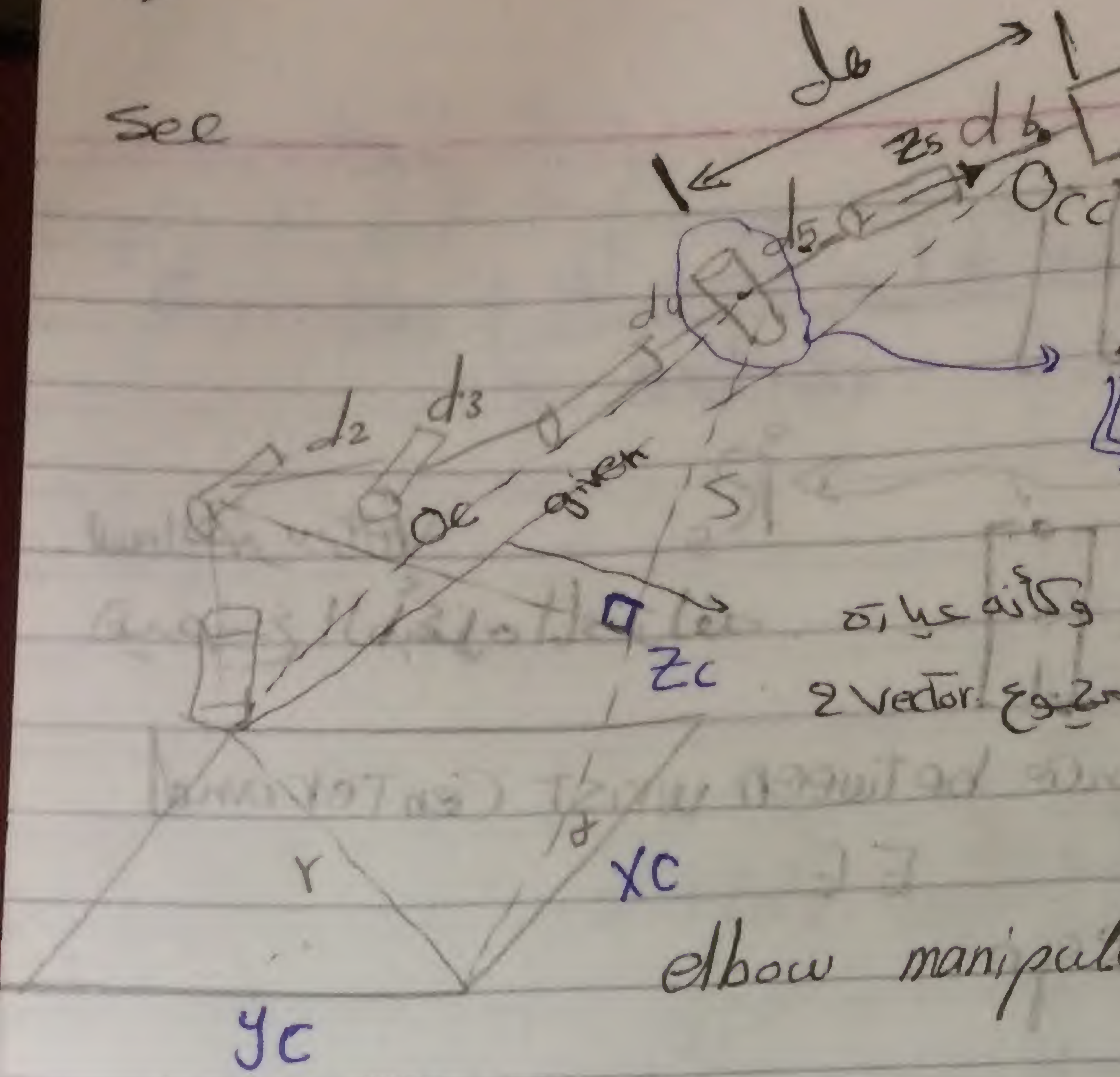
محاضرة ١٣

Robotics

See



ولتحديد 2 زوايا أساسيين أول ٣ زوايا

spherical wrist

"بتقدر بسهولة نحسب أول 3 Angle for 3 first Joint"

وكأنه عبارة عن مجموع 2 vector

elbow manipulator.

closed form :: "Kinematic Decupling"

steps ⟶

* given :- O_{EE} " position of end effector, rotation "

$$R_{EE} \longrightarrow H^{base}_{EE}$$

* Required : q_k " ex: elbow $\Rightarrow \theta_1 \rightarrow \theta_6$ "

* Algorithm :-

1- $O_c = O_{EC}$ "given" $- \; d_6 \, {}^{0}R_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

wrist center.

$$\begin{bmatrix} C30 & -S30 & 0 \\ S30 & C30 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} C20 & -S20 & 0 \\ S20 & C20 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C50 & -S50 & 0 \\ S50 & C50 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$^{0}R_5 \quad R_{EE}$$

$$\Big[\;\;[\;]\;\;[\;\;[\;]\;\Big]$$

$$= \;\longrightarrow\; {}^{0}R_5$$

لانه الصفر لابين Vector

$$O_c = O_{EE} - d_6 R_{EE} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

هيمتنع الأجزاء المختلفه.

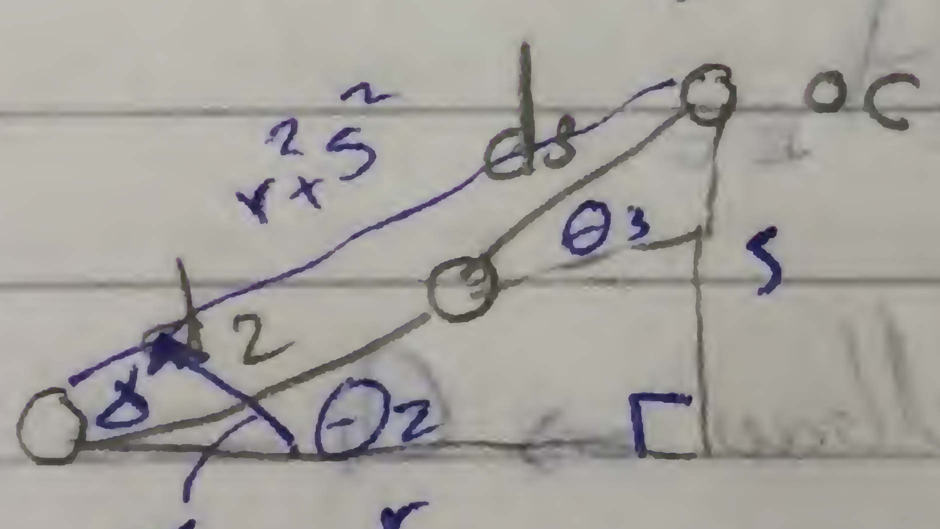
d_6 → distance between wrist Center and EE

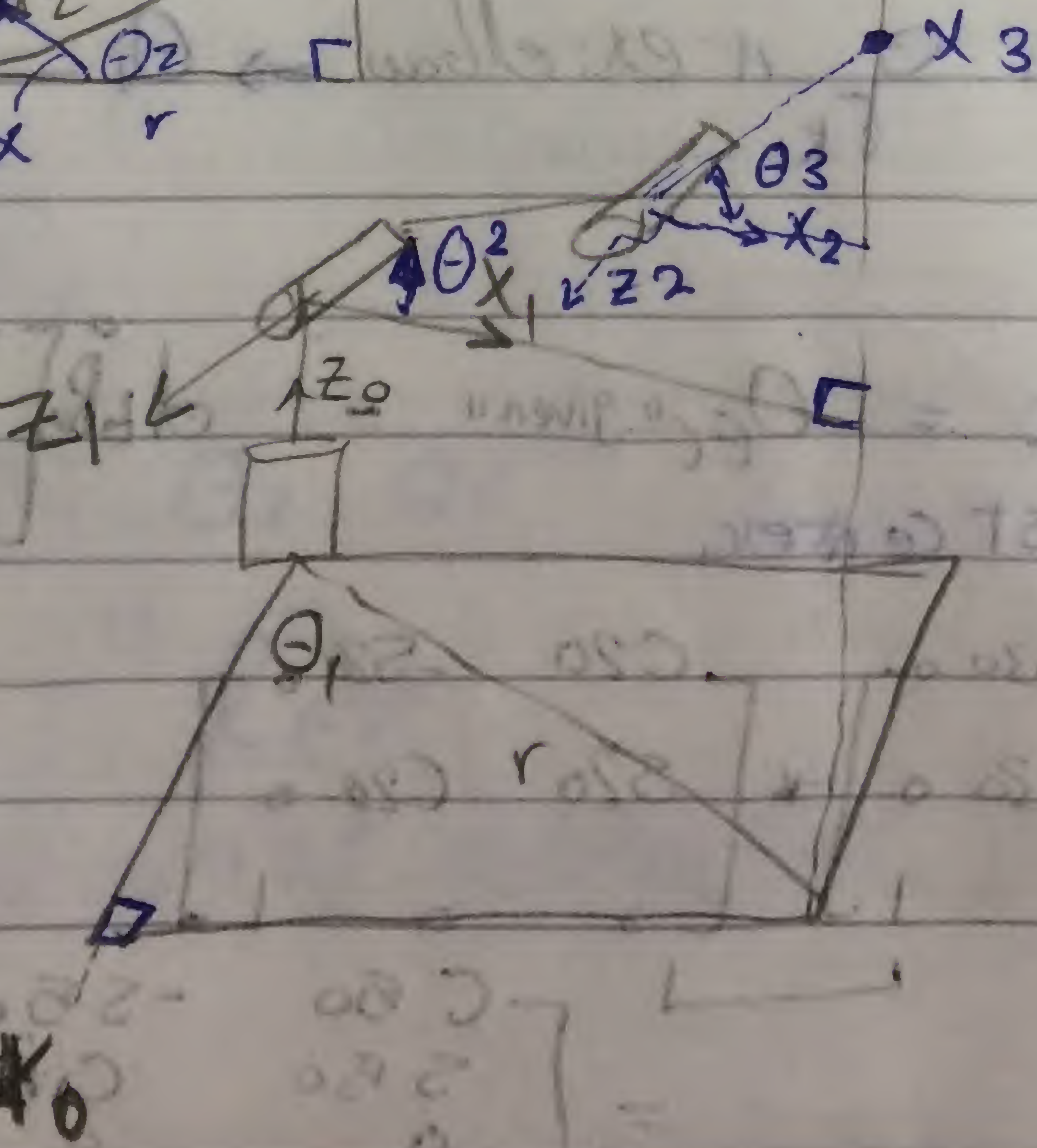
2- From Trigonometric eqn

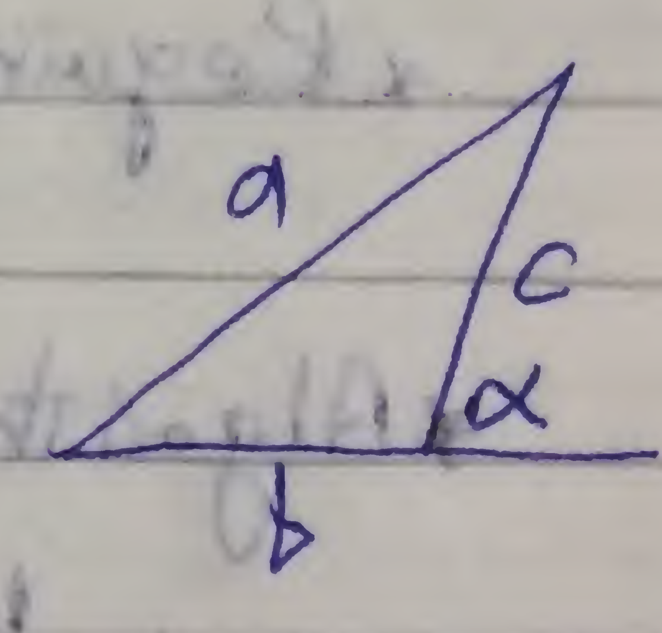
$$\theta_1 = f_1(O_c)$$
$$\theta_2 = f_2(O_c)$$
$$\theta_3 = f_3(O_c)$$

z_c, y_c, x_c لأن مُعلمه معرفة

عند النظر من مستوى موازي للمستوى ذ







$$\cos\alpha = \frac{b^2 + c^2 - a^2}{2bc}$$

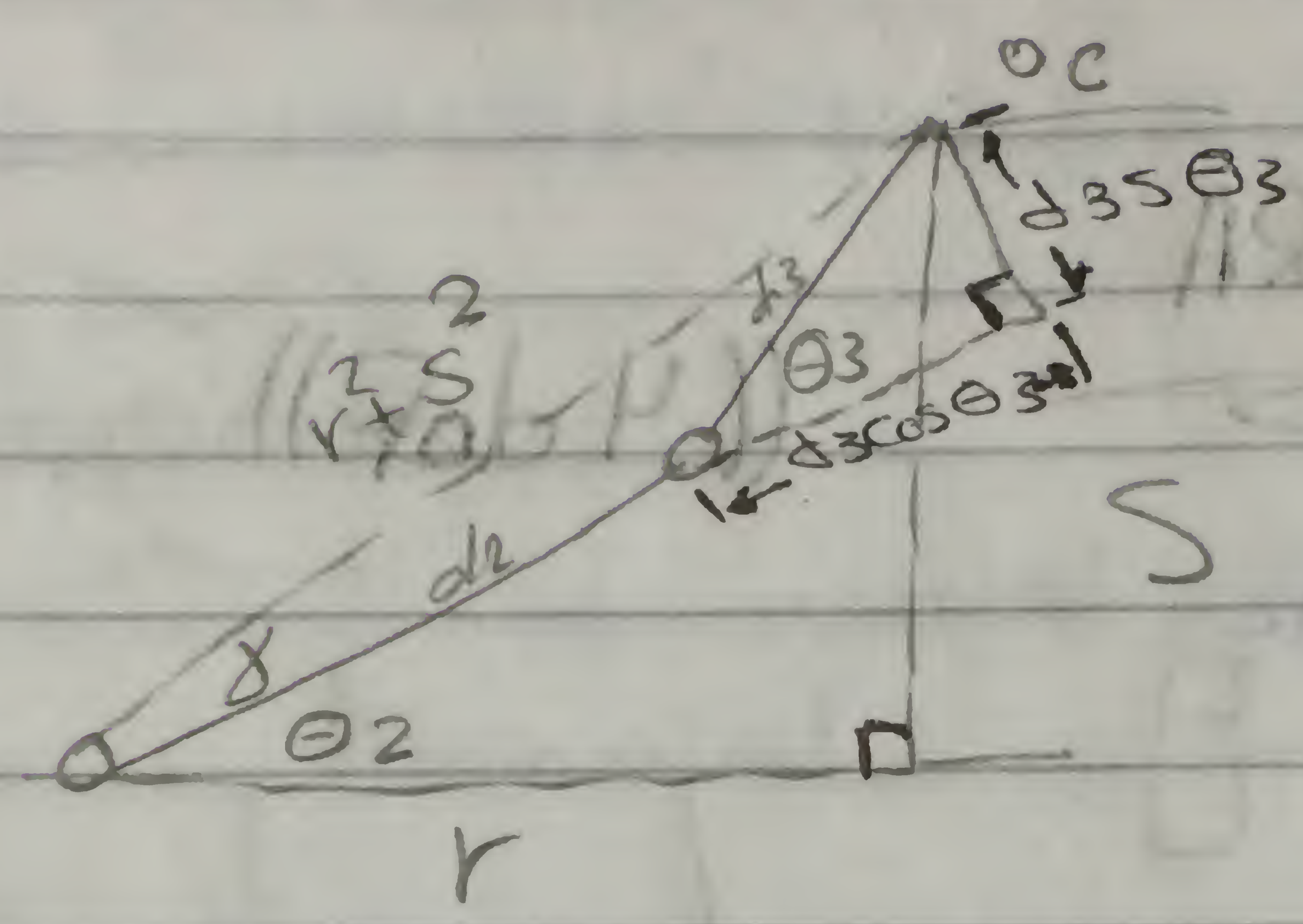
$\theta_1 = \tan^{-1}\left(\frac{y_c}{x_c}\right)$

$$\theta_3 = \cos^{-1}\left(\frac{d_2^2 + d_3^2 - (x_c^2 + y_c^2 + (z_c - d_1)^2)}{2 * d_2 d_3}\right)$$

$\theta_2 =$

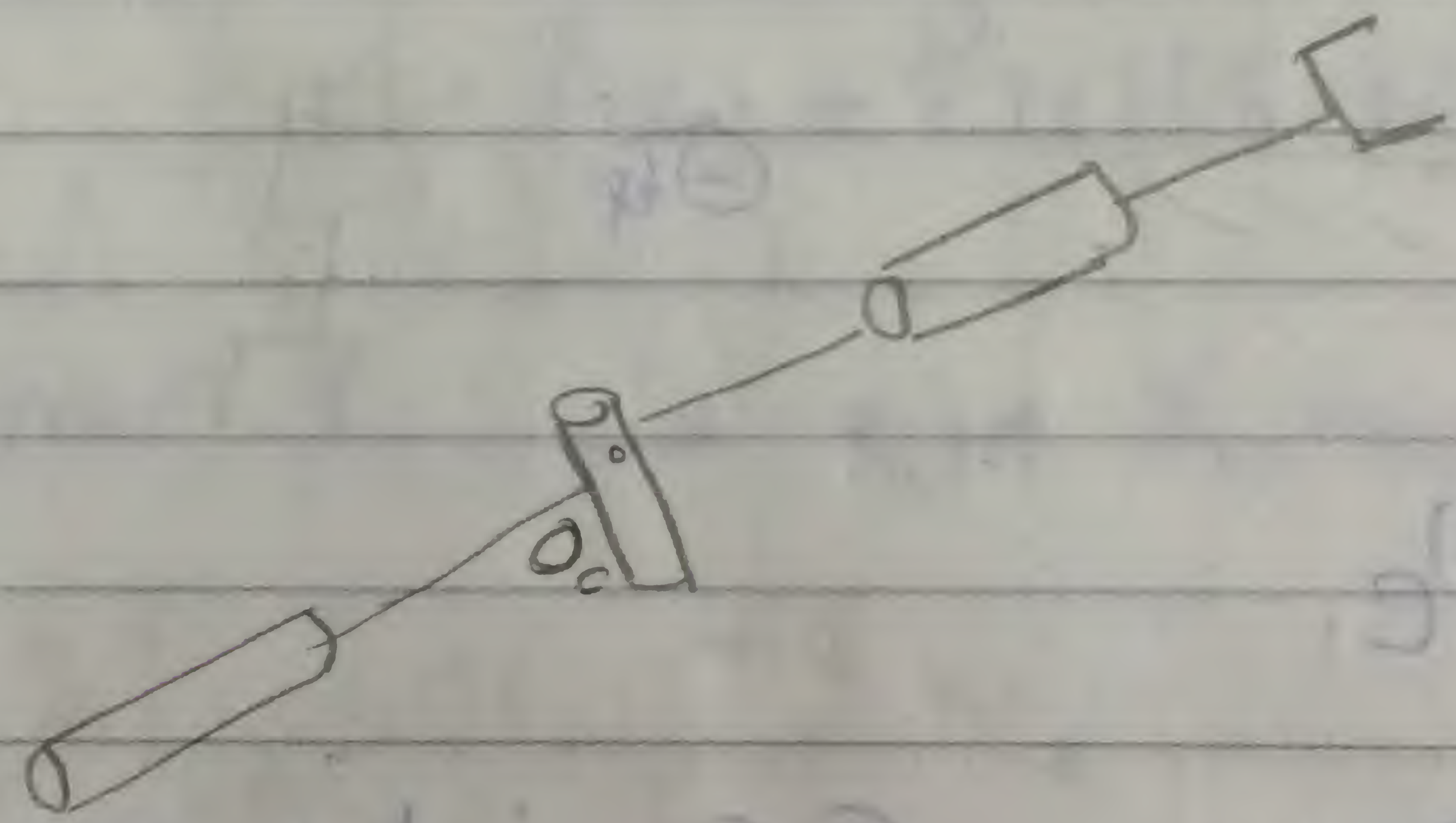
$\alpha = \tan^{-1}\left(\frac{S}{r}\right)$

$\gamma = \tan^{-1}\left(\frac{d_3 S\theta_3}{d_2 + d_3 C\theta_3}\right)$



3- orientation.

desired → θ_4
→ θ_5
→ θ_6



$$R_{EE} = {}^0R_3 \, {}^3R_6$$

${}^0R_3$: from $\theta_1, \theta_2, \theta_3$

${}^3R_6$: $[{}^0R_3]^{-1} R_{EE}$

↳ spherical wrist orientation → Euler Angles.

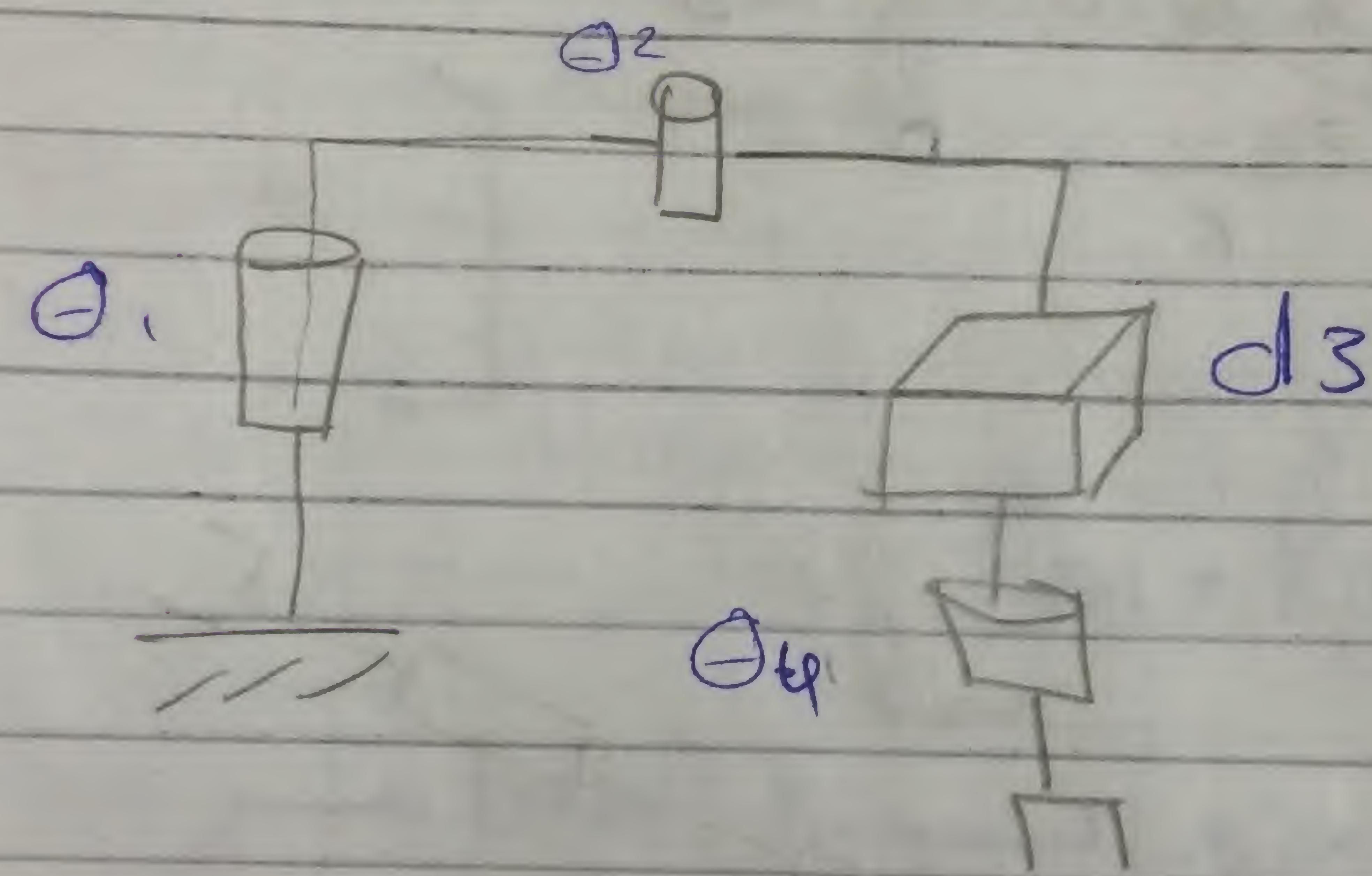
From ${}^3R_6$ using Euler parametrization. $\theta_4, \theta_5, \theta_6$

Euler matrix $\left[\quad \right]$

θ, ϕ, ψ

$\theta_4, \theta_5, \theta_6$

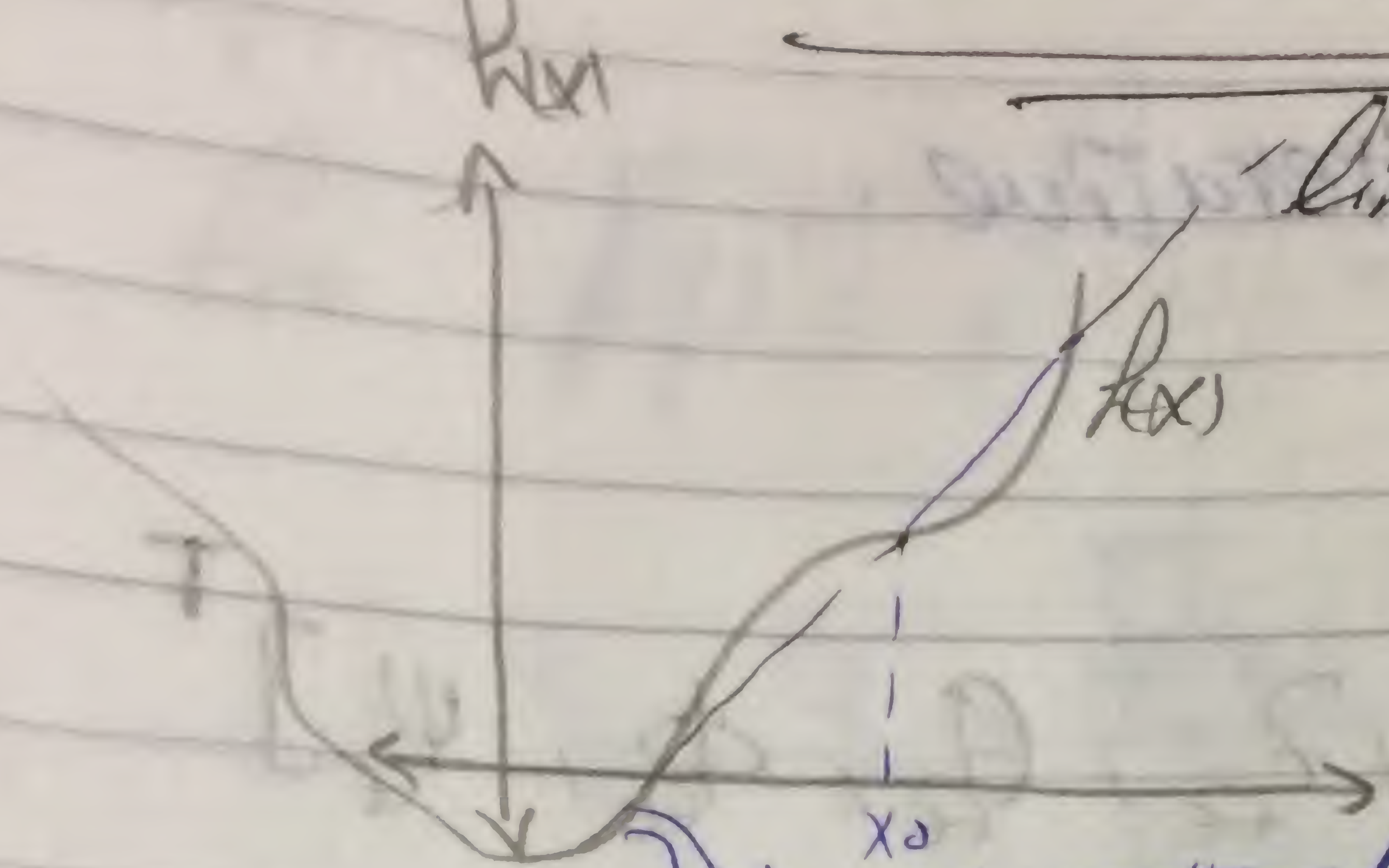
SCARA ((4 dof))



$O_{EE} = O_C$

Position EE indepenton θ_4

Numerical Methods



linear approximation of $f(x)$

التقريب الخطي لـ $f(x)$.

wanted to obtain x that make $f(x)=0$

Tylor exponsion

أى داله نقدر نعبر عنها بمجموعه من binomial.

fns 1 order
fns 2nd order
⋮

$$\underbrace{f(x)}_{\text{linear}} = f(x_0) + f'(x_0)(x - x_0)$$

كل ما x_0 range يزيد يزيد الـ error

كلما بنقرب من x_0 x يكون التعبير أكثر دقة.

$f_x\Big|_{\text{linear}} = 0$ ما

$$0 = f(x_0) + f'(x_0)(\underbrace{x}_{\text{root}} - x_0)$$

$$-f(x_0) = f'(x_0)(x - x_0)$$

$$x - x_0 = -\frac{f(x_0)}{f'(x_0)} = \boxed{x = x_0 - \frac{f(x_0)}{f'(x_0)}}$$

$$\boxed{x = x_0 - \frac{f_{x_0}}{f'_{(x_0)}}}$$ iterative.

given vector $g\ (x_{EE}, y_{EE}, z_{EE}, \theta_{EE}, \phi_{EE}, \psi_{EE}]^T$

desired Pose

N (Dof) robot: $q_i \rightarrow i = 1, \ldots N$

$$f(q_k) = \begin{bmatrix} f_1\ q_k \\ f_2\ q_k \\ \vdots \\ f_6\ q_k \end{bmatrix}$$ from forward Kinematics.

$$\begin{bmatrix} \boxed{\quad\quad} & \vdots \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}_{4\times4}$$ → position of end effector

كلهم ... إيه بيهمني ...

$f(\theta_{EE}, \phi_{EE}, \psi_{EE})$

6 non Linear fns → كل واحد بيتعرف

└ O/P required parameters ←

└ I/P 6 parameter of end effector

g

Steps.

let $F(q) = f(q) - J$ $= 0$ at the desired pose

$$q = q_0 - \frac{F(q_0)}{F'(q_0)}$$

$$\frac{\partial F_1}{\partial q} = \frac{\partial F_1}{\partial q_1} + \frac{\partial F_1}{\partial q_2} + \cdots + \frac{\partial F_1}{\partial q_{rn}}$$

$$\vdots$$

$$\frac{\partial F_n}{\partial q} = \cdots + \frac{\partial F_n}{\partial q_{rn}}$$

multi f^{ns} of multivariable fcou...

$$\frac{\partial f(q)}{\partial q} = \begin{bmatrix} \frac{\partial f_1}{\partial q} \\ \vdots \\ \frac{\partial f_6}{\partial (q)} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial (q_{r1})} + & \cdots & \frac{\partial f_1}{\partial (q_{rn})} \\ \vdots & & \vdots \\ \frac{\partial f_r}{\partial (q_{rn})} + & \cdots & + \frac{\partial f_r}{\partial (q_{rn})} \end{bmatrix} \begin{bmatrix} f_1(q) \\ f_2(q) \\ \vdots \\ f_{rn}(q) \end{bmatrix}$$

Jacobian

أكتر من F^{ns} ع متغير واحد

داله واحده في أكتر من متغير

$$q_{r_{n\times 1}} = q_{r_{n\times 1}} - \underbrace{\cancel{(F(q_r))^{-1}}}_{n\times 6} F(q_r)_{6\times 1}$$

$$\cancel{(F(q))^{-1}}$$

only for squar matrix ⟵ inverse

↓

pseudo-inverse

Artificial intelligence

$$q_r \longrightarrow \boxed{\quad\quad} \longrightarrow f(q_r)$$

$q_1, q_2, \cdots q_n$ Forward Kinematics $x_{EE}, y_{EE}, z_{EE}, \theta_{EE}, \phi_{EE}, \psi$

pose1

⋮

Try ranges for q_r

1- drive forward Kinematics.

2- Try ranges of q and we obtain same Pose

" dataset of pose "

3- Supervised learning "PSO"

list of poses → NN ANFIS → labeled output list of q

Polynomial Regrision

supervised learning

* we have no direct dataset

"Trained $\underline{NN}$"

↳ neural network

desired pose → Trained NN. → required $\underline{q}$